

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA



SISTEMAS DE GESTIÓN DE BASES DE DATOS

Trabajo Práctico N°1: Gestores de Base de Datos

Equipo de Cátedra:

- Profesor: I.S.I. Andrés Pablo Fantín
- J.T.P.: I.S.I. Juan Carlos Fernández
- Auxiliar Adscripta: Luciana Campestrini

Alumnos:

- Cristaldo, Cristian
- Maurel Garcete, Philippe
- Orrego Nilson
- Piragine, Tomás
- Fernandez Bruno Ulises

Grupo 8

Ciclo Lectivo 2024

a) **Prueba de la MV:** enciéndela, inicie sesión, liste el contenido del directorio actual y luego apáguela;

• **Conexión e inicio de sesión (desde la terminal de comandos del host):**

- `multipass shell mv-sgbd`

```
Selecc[3] Seleccionar ubuntu@mv-sgbd: ~
C:\Users\criss>multipass shell mv-sgbd
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-118-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of sáb 17 ago 2024 09:44:17 -03

System load:  0.62           Processes:            136
Usage of /:   10.6% of 19.20GB Users logged in:          0
Memory usage: 12%           IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 6 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Wed Aug  7 22:09:16 2024 from 10.0.2.2
ubuntu@mv-sgbd:~$
```

- **Comandos en la terminal de la máquina virtual:**
 - **Listar contenido del directorio actual:** `ls -al`

```
root@082c0a7ee39b:/# ls -al
total 68
drwxr-xr-x  1 root root 4096 Aug 17 12:36 .
drwxr-xr-x  1 root root 4096 Aug 17 12:36 ..
-rwxr-xr-x  1 root root    0 Aug 17 12:36 .dockerenv
lrwxrwxrwx  1 root root    7 Jun 27 14:20 bin -> usr/bin
drwxr-xr-x  2 root root 4096 Apr 18  2022 boot
drwxr-xr-x  5 root root  360 Aug 17 12:36 dev
drwxr-xr-x  1 root root 4096 Aug 17 12:37 etc
drwxr-xr-x  2 root root 4096 Apr 18  2022 home
lrwxrwxrwx  1 root root    7 Jun 27 14:20 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Jun 27 14:20 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Jun 27 14:20 lib64 -> usr/lib64
lrwxrwxrwx  1 root root   10 Jun 27 14:20 libx32 -> usr/libx32
drwxr-xr-x  2 root root 4096 Jun 27 14:20 media
drwxr-xr-x  2 root root 4096 Jun 27 14:20 mnt
drwxr-xr-x  2 root root 4096 Jun 27 14:20 opt
dr-xr-xr-x 344 root root    0 Aug 17 12:36 proc
drwx----- 2 root root 4096 Jun 27 14:24 root
drwxr-xr-x  1 root root 4096 Aug 17 12:37 run
lrwxrwxrwx  1 root root    8 Jun 27 14:20/sbin -> usr/sbin
drwxr-xr-x  2 root root 4096 Jun 27 14:20 srv
dr-xr-xr-x 13 root root    0 Aug 17 12:36 sys
drwxrwxrwt  1 root root 4096 Aug 17 12:37 tmp
drwxr-xr-x  1 root root 4096 Jun 27 14:20 usr
drwxr-xr-x  1 root root 4096 Jun 27 14:24 var
```

- Comprobar la dirección IP: ip address

```

Seleccionar ubuntu@mv-sgbd:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:da:dd:70 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86385sec preferred_lft 86385sec
    inet6 fe80::5054:ff:feda:dd70/64 scope link
        valid_lft forever preferred_lft forever
ubuntu@mv-sgbd:~$

```

- Apagar equipo: sudo poweroff

```

C:\Users\criss>
ubuntu@mv-sgbd:~$ sudo poweroff

```

c) Realice la instalación del SGBD en la máquina virtual de acuerdo a las indicaciones del manual provista en los enlaces dados a continuación. Copie y pegue cada una de las pantallas que obtiene durante el proceso de instalación y explique con sus palabras lo que se ve en ellas. Asimismo comente cualquier cambio que haya realizado a las opciones cuando no haya elegido dejar la opción predeterminada.

- Agregamos el Repositorio MySQL APT

```

ubuntu@primary:~$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.32-1_all.deb
--2024-08-13 23:17:31-- https://dev.mysql.com/get/mysql-apt-config_0.8.32-1_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 23.222.179.88, 2600:1419:1200:492::2e31, 2600:1419:1200:493::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.222.179.88|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com/mysql-apt-config_0.8.32-1_all.deb [following]
--2024-08-13 23:17:32-- https://repo.mysql.com/mysql-apt-config_0.8.32-1_all.deb
Resolving repo.mysql.com (repo.mysql.com)... 23.55.75.162, 2800:2d20:200a:195::1d68, 2800:2d20:200a:190::1d68
Connecting to repo.mysql.com (repo.mysql.com)|23.55.75.162|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18068 (18K) [application/x-debian-package]
Saving to: 'mysql-apt-config_0.8.32-1_all.deb'

mysql-apt-config_0.8.32-1_all 100%[=====>] 17.64K --.-KB/s in 0.001s

2024-08-13 23:17:32 (29.8 MB/s) - 'mysql-apt-config_0.8.32-1_all.deb' saved [18068/18068]

```

- Instalamos el paquete de lanzamiento descargado

```

ubuntu@primary:~$ sudo dpkg -i mysql-apt-config_0.8.32-1_all.deb
Selecting previously unselected package mysql-apt-config.
(Reading database ... 71867 files and directories currently installed.)
Preparing to unpack mysql-apt-config_0.8.32-1_all.deb ...
Unpacking mysql-apt-config (0.8.32-1) ...
Setting up mysql-apt-config (0.8.32-1) ...

```

- Actualizamos la información del paquete desde el repositorio MySQL APT

```
ubuntu@primary:~$ sudo apt-get update
Get:1 http://repo.mysql.com/apt/ubuntu noble InRelease [22.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 http://repo.mysql.com/apt/ubuntu noble/mysql-8.4-lts Sources [966 B]
Get:7 http://repo.mysql.com/apt/ubuntu noble/mysql-8.4-lts amd64 Packages [17.0 kB]
Get:8 http://repo.mysql.com/apt/ubuntu noble/mysql-tools amd64 Packages [4161 B]
Fetched 44.8 kB in 5s (8679 B/s)
Reading package lists... Done
```

- Instalamos MySQL

```
ubuntu@primary:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmecab2 mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client mysql-common mysql-community-client
  mysql-community-client-core mysql-community-client-plugins mysql-community-server mysql-community-server-core
The following NEW packages will be installed:
  libmecab2 mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client mysql-common mysql-community-client
  mysql-community-client-core mysql-community-client-plugins mysql-community-server mysql-community-server-core
  mysql-server
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 36.6 MB of archives.
After this operation, 262 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libmecab2 amd64 0.996-14ubuntu4 [201 kB]
Get:2 http://repo.mysql.com/apt/ubuntu noble/mysql-8.4-lts amd64 mysql-common amd64 8.4.2-1ubuntu24.04 [67.3 kB]
Get:3 http://repo.mysql.com/apt/ubuntu noble/mysql-8.4-lts amd64 mysql-community-client-plugins amd64 8.4.2-1ubuntu24.04
[1468 kB]
```

- Verificamos la versión de MySQL

```
ubuntu@primary:~$ mysql --version
mysql Ver 8.4.2 for Linux on x86_64 (MySQL Community Server - GPL)
```

d) Puesta a punto del SGBD: Una vez instalado MySQL Server es necesario configurarlo para poder conectarse remotamente creando un usuario que se utilizará en vez del administrador predeterminado (root). Para ello realizar los siguientes pasos:

- Desde el terminal de la MV iniciar una sesión con el cliente mysql (con el usuario root y la contraseña definida durante la instalación):

```
shell> mysql -u root -p Enter password: *****
```

- En la consola de mysql crear un nuevo usuario que pueda conectarse desde cualquier host y asignarle todos los permisos (Importante: esta cuenta es la que deberá usar de ahora en más para trabajar con el SGBD. NO deberá usar más la de 'root').

```

root@082c0a7ee39b:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user 'philippe'@'%' identified by 'sgbd2024'
-> ;
Query OK, 0 rows affected (0.02 sec)

mysql> grant all on *.* to 'philippe'@'%' with grant option;
Query OK, 0 rows affected (0.03 sec)

```

e) Realice los pasos del tutorial disponible en el capítulo 3 del manual de referencia:

5.3 Creating and Using a Database

- Verifica las bases de datos existentes en el servidor.

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

```

- Selecciona la base de datos mysql para comenzar a usarla.

```

mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>

```

5.3.1 Creating and Selecting a Database

- Crear la base de datos.

```

mysql> create database menagerie;
Query OK, 1 row affected (0.01 sec)

mysql>

```

- Seleccionar la base de datos al iniciar MySQL.

```
root@082c0a7ee39b:/# mysql -u philippe -p menagerie
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
Empty set (0.00 sec)
```

5.3.2 Creating a Table

- Verificar si la base de datos está vacía.

```
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

- Crear la tabla.

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
Query OK, 0 rows affected (0.03 sec)
```

- Verificar que la tabla fue creada correctamente.

```
mysql> show tables;
+-----+
| Tables_in_menagerie |
+-----+
| pet                  |
+-----+
1 row in set (0.00 sec)
```

- Describir la estructura de la tabla para confirmar su configuración.

```
mysql> DESCRIBE pet;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(20) | YES  |     | NULL    |       |
| owner | varchar(20) | YES  |     | NULL    |       |
| species | varchar(20) | YES  |     | NULL    |       |
| sex   | char(1)    | YES  |     | NULL    |       |
| birth | date       | YES  |     | NULL    |       |
| death | date       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

5.3.3 Loading Data into a Table

Después de crear una tabla en MySQL, necesitas llenarla con datos. Hay dos formas comunes de hacerlo: usando la instrucción LOAD DATA para cargar un archivo de texto con múltiples registros, o usando INSERT para agregar registros uno por uno.

- Cargar Datos desde un Archivo de Texto (LOAD DATA).

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> LOAD DATA LOCAL INFILE 'C:\Users\criss\Ubuntu_Linux\pet.txt' INTO TABLE pet LINES TERMINATED BY '\r\n';
ERROR 2 (HY000): File 'C:\Users\criss\Ubuntu_Linux\pet.txt' not found (OS errno 2 - No existe el archivo o el directorio)
mysql> LOAD DATA LOCAL INFILE '/home/ubuntu/pet.txt'
-> INTO TABLE pet
-> LINES TERMINATED BY '\r\n';
Query OK, 8 rows affected (0,01 sec)
Records: 8 Deleted: 0 Skipped: 0 Warnings: 0
```

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL

8 rows in set (0,00 sec)

- Insertar Registros Individuales (INSERT)

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> INSERT INTO pet
-> VALUES ('Puffball','Diane','h
Query OK, 1 row affected (0,01 sec)
```

```
mysql> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

9 rows in set (0,00 sec)

5.3.4 Retrieving Information from a Table

5.3.4.1 Selecting All Data

- Seleccionar Todos los Datos de una Tabla

```
mysql> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

```
9 rows in set (0,00 sec)
```

- Borra todos los registros de la tabla:

```
mysql> DELETE FROM pet;
```

Query OK, 9 rows affected (0,02 sec)

- Recarga los datos desde el archivo corregido:

```
mysql> LOAD DATA LOCAL INFILE 'pet.txt' INTO TABLE pet;
```

Query OK, 8 rows affected, 8 warnings (0,02 sec)
Records: 8 Deleted: 0 Skipped: 0 Warnings: 8

- Para corregir sólo el registro de Bowser, puedes usar una instrucción UPDATE:

```
mysql> UPDATE pet SET birth = '1989-08-31' WHERE name = 'Bowser';
```

Query OK, 1 row affected (0,02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

5.3.4.2 Selecting Particular Rows

- Seleccionar Filas Específicas por Nombre:


```

mysql> SELECT * FROM pet WHERE name = 'Bowser';
+-----+-----+-----+-----+-----+-----+
| name  | owner | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Bowser | Diane | dog     | m    | 1979-08-31 | 1995-07-29 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)

```

- Filtrar por Fecha:

```

mysql> SELECT * FROM pet WHERE birth >= '1998-1-1';
+-----+-----+-----+-----+-----+-----+
| name    | owner | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Chirpy  | Gwen  | bird    | f    | 1998-09-11 | NULL       |
| Puffball | Diane | hamster | f    | 1999-03-30 | NULL       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)

```

- Usa el operador AND para combinar múltiples condiciones.

```

mysql> SELECT * FROM pet WHERE species = 'dog' AND sex = 'f';
+-----+-----+-----+-----+-----+-----+
| name  | owner | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Buffy | Harold | dog     | f    | 1989-05-13 | NULL       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)

```

- Usa el operador OR para seleccionar filas que cumplan al menos una de varias condiciones.

```

mysql> SELECT * FROM pet WHERE species = 'snake' OR species = 'bird';
+-----+-----+-----+-----+-----+-----+
| name    | owner | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Chirpy  | Gwen  | bird    | f    | 1998-09-11 | NULL       |
| Whistler | Gwen  | bird    | NULL | 1997-12-09 | NULL       |
| Slim    | Benny | snake   | m    | 1996-04-29 | NULL       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

```

- Combina los operadores AND y OR con paréntesis para controlar el orden de evaluación.

```

mysql> SELECT * FROM pet WHERE (species = 'cat' AND sex = 'm')
-> OR (species = 'dog' AND sex = 'f');
+-----+-----+-----+-----+-----+-----+
| name  | owner | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Claws | Gwen  | cat     | m    | 1994-03-17 | NULL       |
| Buffy | Harold | dog     | f    | 1989-05-13 | NULL       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)

```

5.3.4.3 Selecting Particular Columns

- Seleccionar Columnas Específicas:

```
Seleccionar ubuntu@mv-sgbd: ~  
mysql> SELECT name, birth FROM pet;  
+-----+-----+  
| name   | birth   |  
+-----+-----+  
| Fluffy | 1993-02-04 |  
| Claws  | 1994-03-17 |  
| Buffy  | 1989-05-13 |  
| Fang   | 1990-08-27 |  
| Bowser | 1979-08-31 |  
| Chirpy | 1998-09-11 |  
| Whistler | 1997-12-09 |  
| Slim   | 1996-04-29 |  
| Puffball | 1999-03-30 |  
+-----+-----+  
9 rows in set (0,00 sec)
```

- Seleccionar una Columna Única:

```
Seleccionar ubuntu@mv-sgbd: ~  
mysql> SELECT owner FROM pet;  
+-----+  
| owner |  
+-----+  
| Harold |  
| Gwen  |  
| Harold |  
| Benny |  
| Diane |  
| Gwen  |  
| Gwen  |  
| Benny |  
| Diane |  
+-----+  
9 rows in set (0,00 sec)
```

- Eliminar Duplicados con DISTINCT:

```
Seleccionar ubuntu@mv-sgbd: ~  
mysql> SELECT DISTINCT owner FROM pet;  
+-----+  
| owner |  
+-----+  
| Harold |  
| Gwen  |  
| Benny |  
| Diane |  
+-----+  
4 rows in set (0,00 sec)
```

- Combinar Selección de Columnas con Filtros:

```
mysql> SELECT name, species, birth FROM pet
-> WHERE species = 'dog' OR species = 'cat';
```

name	species	birth
Fluffy	cat	1993-02-04
Claws	cat	1994-03-17
Buffy	dog	1989-05-13
Fang	dog	1990-08-27
Bowser	dog	1979-08-31

5 rows in set (0,00 sec)

5.3.4.4 Sorting Rows

- Ordenar Resultados por Columna:

```
mysql> SELECT name, birth FROM pet ORDER BY birth;
```

name	birth
Bowser	1979-08-31
Buffy	1989-05-13
Fang	1990-08-27
Fluffy	1993-02-04
Claws	1994-03-17
Slim	1996-04-29
Whistler	1997-12-09
Chirpy	1998-09-11
Puffball	1999-03-30

9 rows in set (0,00 sec)

- Ordenar de Forma Descendente:

```
mysql> SELECT name, birth FROM pet ORDER BY birth DESC;
```

name	birth
Puffball	1999-03-30
Chirpy	1998-09-11
Whistler	1997-12-09
Slim	1996-04-29
Claws	1994-03-17
Fluffy	1993-02-04
Fang	1990-08-27
Buffy	1989-05-13
Bowser	1979-08-31

9 rows in set (0,00 sec)

- Ordenar por Múltiples Columnas:

```
mysql> SELECT name, species, birth FROM pet
-> ORDER BY species, birth DESC;
```

name	species	birth
Chirpy	bird	1998-09-11
Whistler	bird	1997-12-09
Claws	cat	1994-03-17
Fluffy	cat	1993-02-04
Fang	dog	1990-08-27
Buffy	dog	1989-05-13
Bowser	dog	1979-08-31
Puffball	hamster	1999-03-30
Slim	snake	1996-04-29

9 rows in set (0,00 sec)

5.3.4.5 Date Calculations

- Calcular Edad de Animales: la función `TIMESTAMPDIFF()` para calcular la diferencia entre dos fechas. Puedes obtener la edad de tus mascotas en años.

```
mysql> SELECT name, birth, CURDATE(),
-> TIMESTAMPDIFF(YEAR,birth,CURDATE()) AS age
-> FROM pet;
```

name	birth	CURDATE()	age
Fluffy	1993-02-04	2024-08-18	31
Claws	1994-03-17	2024-08-18	30
Buffy	1989-05-13	2024-08-18	35
Fang	1990-08-27	2024-08-18	33
Bowser	1979-08-31	2024-08-18	44
Chirpy	1998-09-11	2024-08-18	25
Whistler	1997-12-09	2024-08-18	26
Slim	1996-04-29	2024-08-18	28
Puffball	1999-03-30	2024-08-18	25

9 rows in set (0,00 sec)

Resultado ordenado por nombre:

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> SELECT name, birth, CURDATE(),
->         TIMESTAMPDIF(YEAR,birth,CURDATE()) AS age
->         FROM pet ORDER BY name;
+-----+-----+-----+-----+
| name   | birth   | CURDATE() | age |
+-----+-----+-----+-----+
| Bowser  | 1979-08-31 | 2024-08-18 | 44 |
| Buffy   | 1989-05-13 | 2024-08-18 | 35 |
| Chirpy  | 1998-09-11 | 2024-08-18 | 25 |
| Claws   | 1994-03-17 | 2024-08-18 | 30 |
| Fang    | 1990-08-27 | 2024-08-18 | 33 |
| Fluffy  | 1993-02-04 | 2024-08-18 | 31 |
| Puffball | 1999-03-30 | 2024-08-18 | 25 |
| Slim    | 1996-04-29 | 2024-08-18 | 28 |
| Whistler | 1997-12-09 | 2024-08-18 | 26 |
+-----+-----+-----+-----+
9 rows in set (0,00 sec)
```

- Ordenar por Edad:

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> SELECT name, birth, CURDATE(),
->         TIMESTAMPDIF(YEAR,birth,CURDATE()) AS age
->         FROM pet ORDER BY age;
+-----+-----+-----+-----+
| name   | birth   | CURDATE() | age |
+-----+-----+-----+-----+
| Chirpy  | 1998-09-11 | 2024-08-18 | 25 |
| Puffball | 1999-03-30 | 2024-08-18 | 25 |
| Whistler | 1997-12-09 | 2024-08-18 | 26 |
| Slim    | 1996-04-29 | 2024-08-18 | 28 |
| Claws   | 1994-03-17 | 2024-08-18 | 30 |
| Fluffy  | 1993-02-04 | 2024-08-18 | 31 |
| Fang    | 1990-08-27 | 2024-08-18 | 33 |
| Buffy   | 1989-05-13 | 2024-08-18 | 35 |
| Bowser  | 1979-08-31 | 2024-08-18 | 44 |
+-----+-----+-----+-----+
9 rows in set (0,00 sec)
```

- Edad en la Muerte: solo considera las mascotas cuya fecha de muerte no sea NULL.

```
Seleccionar ubuntu@mv-sgbd: ~
mysql> SELECT name, birth, death,
->         TIMESTAMPDIF(YEAR,birth,death) AS age
->         FROM pet WHERE death IS NOT NULL ORDER BY age;
+-----+-----+-----+-----+
| name   | birth   | death   | age |
+-----+-----+-----+-----+
| Bowser  | 1979-08-31 | 1995-07-29 | 15 |
+-----+-----+-----+-----+
1 row in set (0,00 sec)
```

- Encontrar Cumpleaños del Próximo Mes: para obtener las fechas de cumpleaños del próximo mes, usar MONTH() y DATE_ADD().

```
mysql> SELECT name, birth, MONTH(birth) FROM pet;
```

name	birth	MONTH(birth)
Fluffy	1993-02-04	2
Claws	1994-03-17	3
Buffy	1989-05-13	5
Fang	1990-08-27	8
Bowser	1979-08-31	8
Chirpy	1998-09-11	9
Whistler	1997-12-09	12
Slim	1996-04-29	4
Puffball	1999-03-30	3

9 rows in set (0,00 sec)

```
mysql> SELECT name, birth FROM pet WHERE MONTH(birth) = 5;
```

name	birth
Buffy	1989-05-13

1 row in set (0,00 sec)

```
mysql> SELECT name, birth FROM pet
-> WHERE MONTH(birth) = MONTH(DATE_ADD(CURDATE(),INTERVAL 1 MONTH));
```

name	birth
Chirpy	1998-09-11

1 row in set (0,01 sec)

Si el mes actual es diciembre, usar la función MOD() para evitar problemas con el valor del mes.

```
mysql> SELECT name, birth FROM pet
-> WHERE MONTH(birth) = MOD(MONTH(CURDATE()), 12) + 1;
```

name	birth
Chirpy	1998-09-11

1 row in set (0,00 sec)

- Manejo de Fechas Inválidas:

```
mysql> SELECT '2018-10-31' + INTERVAL 1 DAY;
+-----+
| '2018-10-31' + INTERVAL 1 DAY |
+-----+
| 2018-11-01                      |
+-----+
1 row in set (0,00 sec)
```

```
mysql> SELECT '2018-10-32' + INTERVAL 1 DAY;
+-----+
| '2018-10-32' + INTERVAL 1 DAY |
+-----+
| NULL                          |
+-----+
1 row in set, 1 warning (0,01 sec)
```

Ver advertencias:

```
mysql> SHOW WARNINGS;
+-----+
| Level | Code | Message                                     |
+-----+
| Warning | 1292 | Incorrect datetime value: '2018-10-32' |
+-----+
1 row in set (0,00 sec)
```

5.3.4.6 Working with NULL Values

- Probar NULL: Usa los operadores `IS NULL` y `IS NOT NULL` para comprobar si un valor es NULL o no.

```
mysql> SELECT 1 IS NULL, 1 IS NOT NULL;
+-----+
| 1 IS NULL | 1 IS NOT NULL |
+-----+
| 0         | 1             |
+-----+
1 row in set (0.00 sec)
```

- `1 IS NULL`: Verifica si el valor 1 es NULL. Como 1 no es NULL, el resultado será 0 (falso).
- `1 IS NOT NULL`: Verifica si el valor 1 no es NULL. Como 1 no es NULL, el resultado será 1 (verdadero).

- Evitar Comparaciones Directas: No uses operadores de comparación aritmética (=, <, >, <>) con NULL, ya que siempre devolverán NULL como resultado.

```
mysql> SELECT 1 = NULL, 1 <> NULL, 1 < NULL, 1 > NULL;
+-----+-----+-----+-----+
| 1 = NULL | 1 <> NULL | 1 < NULL | 1 > NULL |
+-----+-----+-----+-----+
|      NULL |      NULL |      NULL |      NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- Filtrado con NULL: Usa IS NOT NULL para filtrar valores NULL en consultas, especialmente para comprobar si una columna tiene un valor.

```
mysql> SELECT * FROM pet WHERE death IS NOT NULL;
+-----+-----+-----+-----+-----+-----+
| name   | owner | species | sex | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Bowser | Diane | dog      | m   | 1979-08-31 | 1995-07-29 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

En este caso, Bowser es la única mascota que tiene fecha de muerte.

- Inserción en Columnas NOT NULL: Es posible insertar un 0 o una cadena vacía en una columna definida como NOT NULL, ya que estos son valores válidos y no se consideran NULL.

```
mysql> SELECT 0 IS NULL, 0 IS NOT NULL, '' IS NULL, '' IS NOT NULL;
+-----+-----+-----+-----+
| 0 IS NULL | 0 IS NOT NULL | '' IS NULL | '' IS NOT NULL |
+-----+-----+-----+-----+
|          0 |          1 |          0 |          1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 0 IS NULL: Verifica si el valor 0 es NULL. Como 0 no es NULL, el resultado será 0 (falso).
- 0 IS NOT NULL: Verifica si el valor 0 no es NULL. Como 0 no es NULL, el resultado será 1 (verdadero).
- '' IS NULL: Verifica si una cadena vacía (") es NULL. Como una cadena vacía no es NULL, el resultado será 0 (falso).
- '' IS NOT NULL: Verifica si una cadena vacía (") no es NULL. Como una cadena vacía no es NULL, el resultado será 1 (verdadero).

5.3.4.7 Pattern Matching

- Encontrar nombres que empiecen con "b":

```
mysql> SELECT * FROM pet WHERE name LIKE 'b%';
```

name	owner	species	sex	birth	death
Buffy	Harold	dog	f	1989-05-13	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29

2 rows in set (0.00 sec)

- Encontrar nombres que terminen con "fy":

```
mysql> SELECT * FROM pet WHERE name LIKE '%fy';
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

2 rows in set (0.00 sec)

- Encontrar nombres que contengan "w":

```
mysql> SELECT * FROM pet WHERE name LIKE '%w%';
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Whistler	Gwen	bird	NULL	1997-12-09	NULL

3 rows in set (0.00 sec)

- Encontrar nombres con exactamente 5 caracteres:

```
mysql> SELECT * FROM pet WHERE name LIKE '_____';
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

2 rows in set (0.00 sec)

El otro tipo de coincidencia de patrones que proporciona MySQL utiliza expresiones regulares extendidas:

- Encontrar nombres que empiecen con "b":

```
mysql> SELECT * FROM pet WHERE REGEXP LIKE(name, '^b');
+-----+-----+-----+-----+-----+-----+
| name   | owner  | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Buffy  | Harold | dog     | f    | 1989-05-13 | NULL       |
| Bowser | Diane  | dog     | m    | 1979-08-31 | 1995-07-29 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.13 sec)
```

^ indica el inicio de la cadena

- Encontrar nombres que terminen con "fy":

```
mysql> SELECT * FROM pet WHERE REGEXP LIKE(name, 'fy$');
+-----+-----+-----+-----+-----+-----+
| name   | owner  | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Fluffy | Harold | cat     | f    | 1993-02-04 | NULL       |
| Buffy  | Harold | dog     | f    | 1989-05-13 | NULL       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

\$ indica el final de la cadena.

- Encontrar nombres que contengan "w":

```
mysql> SELECT * FROM pet WHERE REGEXP LIKE(name, 'w');
+-----+-----+-----+-----+-----+-----+
| name   | owner  | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Claws  | Gwen   | cat     | m    | 1994-03-17 | NULL       |
| Bowser | Diane  | dog     | m    | 1979-08-31 | 1995-07-29 |
| Whistler | Gwen  | bird    | NULL | 1997-12-09 | NULL       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

- Encontrar nombres con exactamente 5 caracteres:

```
mysql> SELECT * FROM pet WHERE REGEXP LIKE(name, '^.....$');
+-----+-----+-----+-----+-----+-----+
| name   | owner  | species | sex  | birth      | death      |
+-----+-----+-----+-----+-----+-----+
| Claws  | Gwen   | cat     | m    | 1994-03-17 | NULL       |
| Buffy  | Harold | dog     | f    | 1989-05-13 | NULL       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

. representa cualquier carácter.

- También se puede escribir la consulta anterior utilizando el operador {n} ("repetir-n-veces"):

```
mysql> SELECT * FROM pet WHERE REGEXP LIKE(name, '^.{5}$');
+-----+-----+-----+-----+-----+-----+
| name | owner | species | sex | birth      | death |
+-----+-----+-----+-----+-----+-----+
| Claws | Gwen  | cat     | m   | 1994-03-17 | NULL   |
| Buffy | Harold | dog     | f   | 1989-05-13 | NULL   |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5.3.4.8 Counting Rows

- Contar la cantidad de mascotas

```
mysql> SELECT COUNT(*) FROM pet;
+-----+
| COUNT(*) |
+-----+
|          8 |
+-----+
1 row in set (0.01 sec)
```

- Averiguar cuántas mascotas tiene cada dueño

```
mysql> SELECT owner, COUNT(*) FROM pet GROUP BY owner;
+-----+-----+
| owner | COUNT(*) |
+-----+-----+
| Harold |          2 |
| Gwen  |          3 |
| Benny |          2 |
| Diane |          1 |
+-----+-----+
4 rows in set (0.03 sec)
```

- Averiguar cantidad de animales por especie

```
mysql> SELECT species, COUNT(*) FROM pet GROUP BY species
-> ;
+-----+-----+
| species | COUNT(*) |
+-----+-----+
| cat     |          2 |
| dog     |          3 |
| bird    |          2 |
| snake   |          1 |
+-----+-----+
4 rows in set (0.00 sec)
```

- Averiguar cantidad de animales por sexo

```
mysql> SELECT sex, COUNT(*) FROM pet GROUP BY sex;
```

sex	COUNT(*)
f	3
m	4
NULL	1

3 rows in set (0.00 sec)

NULL indica que el sexo es desconocido.

- Cantidad de animales combinados por especies y sexo:

```
mysql> SELECT species, sex, COUNT(*) FROM pet GROUP BY species, sex;
```

species	sex	COUNT(*)
cat	f	1
cat	m	1
dog	f	1
dog	m	2
bird	f	1
bird	NULL	1
snake	m	1

7 rows in set (0.00 sec)

- Si quisiéramos saber solo la cantidad de perros y gatos:

```
mysql> SELECT species, sex, COUNT(*) FROM pet
-> WHERE species = 'dog' OR species = 'cat'
-> GROUP BY species, sex;
```

species	sex	COUNT(*)
cat	f	1
cat	m	1
dog	f	1
dog	m	2

4 rows in set (0.01 sec)

- Si se quisiera conocer el número de animales por sexo solo para Animales cuyo sexo se conoce:

```
mysql> SELECT species, sex, COUNT(*) FROM pet
-> WHERE sex IS NOT NULL
-> GROUP BY species, sex;
```

species	sex	COUNT(*)
cat	f	1
cat	m	1
dog	f	1
dog	m	2
bird	f	1
snake	m	1

6 rows in set (0.01 sec)

```
mysql> SET sql_mode = '';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT owner, COUNT(*) FROM pet;
```

owner	COUNT(*)
Harold	8

1 row in set (0.00 sec)

5.3.4.9 Using More Than one Table

- Creamos una tabla para registrar eventos que ocurren durante la vida de las mascotas:

```
mysql> CREATE TABLE event (name VARCHAR(20), date DATE,
-> type VARCHAR(15), remark VARCHAR(255));
Query OK, 0 rows affected (0.06 sec)
```

- Importamos los datos desde el archivo event.txt a la tabla event

```
mysql> LOAD DATA LOCAL INFILE '/home/ubuntu/menagerie-db/event.txt' INTO TABLE event FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n' (name, date, type, remark);
Query OK, 10 rows affected, 2 warnings (0.18 sec)
Records: 10 Deleted: 0 Skipped: 0 Warnings: 2

mysql> select * from event
-> ;
+-----+-----+-----+-----+
| name | date | type | remark |
+-----+-----+-----+-----+
| Fluffy | 1995-05-15 | litter | 4 kittens, 3 female, 1 male |
| Buffy | 1993-06-23 | litter | 5 puppies, 2 female, 3 male |
| Buffy | 1994-06-19 | litter | 3 puppies, 3 female |
| Chirpy | 1999-03-21 | vet | needed beak straightened |
| Slim | 1997-08-03 | vet | broken rib |
| Bowser | 1991-10-12 | kennel | NULL |
| Fang | 1991-10-12 | kennel | NULL |
| Fang | 1998-08-28 | birthday | Gave him a new chew toy |
| Claws | 1998-03-17 | birthday | Gave him a new flea collar |
| Whistler | 1998-12-09 | birthday | First birthday |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- Si quisiéramos averiguar la edad a la que cada mascota tiene su camada:

```
mysql> SELECT pet.name,
->         TIMESTAMPDIFF(YEAR,birth,date) AS age,
->         remark
->         FROM pet INNER JOIN event
->         ON pet.name = event.name
->         WHERE event.type = 'litter';
+-----+-----+-----+
| name | age | remark |
+-----+-----+-----+
| Fluffy | 2 | 4 kittens, 3 female, 1 male |
| Buffy | 5 | 3 puppies, 3 female |
| Buffy | 4 | 5 puppies, 2 female, 3 male |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Si quisiéramos encontrar parejas reproductoras utilizando solo la tabla mascotas:

```
mysql> SELECT p1.name, p1.sex, p2.name, p2.sex, p1.species
->         FROM pet AS p1 INNER JOIN pet AS p2
->         ON p1.species = p2.species
->         AND p1.sex = 'f' AND p1.death IS NULL
->         AND p2.sex = 'm' AND p2.death IS NULL;
+-----+-----+-----+-----+-----+
| name | sex | name | sex | species |
+-----+-----+-----+-----+-----+
| Fluffy | f | Claws | m | cat |
| Buffy | f | Fang | m | dog |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

f) ¿Qué es una "instancia" en MySQL? ¿Cuál es la diferencia con una "base de datos" o "esquema"?

Instancia

Una **instancia** de MySQL se refiere a una instalación en ejecución del servidor MySQL. Es el proceso del sistema de gestión de bases de datos que maneja la conexión de clientes, ejecuta consultas, y gestiona las bases de datos y la información almacenada. Básicamente, una instancia es el motor MySQL en ejecución que está trabajando con datos y respondiendo a las solicitudes.

Base de datos

Una **base de datos** en MySQL es una colección de datos organizados en tablas. Cada base de datos es un contenedor lógico que almacena los datos, sus estructuras (tablas, índices, vistas, procedimientos almacenados, etc.), y las relaciones entre ellos. Una base de datos puede contener múltiples tablas y otros objetos de base de datos.

Schema

El término **esquema** en MySQL se usa de manera similar a "base de datos". En MySQL, un esquema es esencialmente lo mismo que una base de datos. Contiene tablas, vistas, índices, y otros objetos relacionados. La diferencia en el uso de términos puede depender del contexto y de otras bases de datos. Por ejemplo, en algunos sistemas de gestión de bases de datos, el esquema puede referirse a una estructura más amplia que incluye la base de datos y su organización lógica.

Diferencias clave

- **Instancia:** Es el proceso del servidor MySQL que maneja todas las operaciones de bases de datos. Puede existir más de una instancia en un mismo servidor, cada una independiente, escuchando en diferentes puertos o sockets. Esto es útil para pruebas, desarrollo, y replicación.
- **Base de Datos/Esquema:** Son contenedores lógicos para los datos y objetos dentro del servidor MySQL. En MySQL, una base de datos y un esquema son esencialmente sinónimos.

g) Enumere y dé una breve descripción sobre clasificaciones y tipos de datos soportados por el gestor (capítulos 11.1 a 11.5 del manual de referencia).

Literal Values:

1. **String Literals:** Un literal de cadena puede especificar un conjunto de caracteres y una intercalación. Las secuencias con una barra invertida () tienen un significado especial a menos que se habilite un modo SQL que deshabilite estas secuencias. Las cadenas entrecomilladas contiguas se concatenan automáticamente.
2. **Numeric Literals:** Incluyen literales de valor exacto (enteros y DECIMAL) y literales de valor aproximado (punto flotante).
3. **Date and Time Literals:** Los valores de fecha y hora pueden representarse en varios formatos, dependiendo del contexto y tipo de valor.
4. **Hexadecimal Literals:** Se escriben utilizando dígitos hexadecimales, con opciones de notación que pueden variar según el caso.
5. **Bit-Value Literals:** Se escriben utilizando una notación específica que indica un valor binario.
6. **Boolean Literals:** Las constantes TRUE y FALSE se evalúan como 1 y 0, respectivamente.
7. **NULL Values:** El valor NULL significa "sin datos" y puede escribirse en cualquier combinación de mayúsculas y minúsculas.

Schema Object Names:

1. **Identifier Length Limits:** Existen límites máximos para la longitud de cada tipo de identificador.
2. **Identifier Qualifiers:** Los nombres de objetos pueden ser calificados o no, según el contexto. Un nombre calificado incluye un calificador para clarificar el contexto, cómo especificar una base de datos particular.
3. **Identifier Case Sensitivity:** La sensibilidad a mayúsculas y minúsculas depende del sistema operativo. Para evitar problemas, se recomienda usar nombres en minúsculas para máxima portabilidad.
4. **Mapping of Identifiers to File Names:** Los identificadores de bases de datos y tablas se corresponden con nombres en el sistema de archivos. La sensibilidad a mayúsculas y minúsculas depende de las características del sistema de archivos.
5. **Function Name Parsing and Resolution:** MySQL soporta funciones nativas, cargables y almacenadas. El servidor determina qué función usar según el contexto.

Keywords and Reserved Words:

- Las palabras clave tienen significado en SQL. Algunas están reservadas y requieren tratamiento especial cuando se usan como identificadores. Las palabras no reservadas se pueden usar como identificadores sin necesidad de comillas, mientras que las palabras reservadas deben citarse para su uso como identificadores. Los nombres de funciones integradas pueden usarse como identificadores, pero requieren cuidado para evitar conflictos con las invocaciones de funciones. Algunas de estas palabras reservadas son:
 - **SELECT**
 - **DELETE**
 - **DATABASE**
 - **AND**

User-Defined Variables:

- Las variables definidas por el usuario en MySQL se utilizan para almacenar y pasar valores entre declaraciones dentro de una misma sesión. Estas variables se escriben como @var_name y son específicas de la sesión, no visibles para otras sesiones. Los nombres no son sensibles a mayúsculas y tienen un límite de 64 caracteres. Se pueden asignar mediante SET, usando = o := como operadores. Pueden almacenar tipos limitados de datos, como enteros, decimales y cadenas, y su tipo se determina al momento de su primera asignación. No se pueden usar como identificadores directos en SQL, pero se pueden utilizar para construir declaraciones dinámicas.

Expressions:

- Las expresiones deben seguir reglas gramaticales específicas en MySQL, y se pueden utilizar diversos tipos de términos dentro de estas expresiones.
 - **Estructura de Expresiones:** Las expresiones en MySQL pueden incluir operadores lógicos (OR, AND, NOT), operadores de comparación (=, >, <, etc.) y operaciones aritméticas (+, -, *, /).
 - **Expresiones Temporales:** Se utilizan intervalos temporales en funciones como DATE_ADD y DATE_SUB para realizar operaciones con fechas. Ejemplos de unidades temporales son DAY, HOUR, YEAR, y combinaciones como DAY_HOUR.
 - **Formato de Intervalos:** Los intervalos se especifican como INTERVAL expr unidad, donde expr es la cantidad y unidad es la medida de tiempo. Por ejemplo, INTERVAL 1 DAY añade un día a una fecha.
 - **Consideraciones:** Al operar con fechas, si el intervalo incluye partes de tiempo, el resultado es un valor datetime. También, si se suma un mes a una fecha que no existe en el mes resultante, MySQL ajusta el día al máximo permitido en ese mes.

h) ¿Qué son los “storage engines” en MySQL? Elabore un concepto y describa brevemente al menos tres de ellos.

Un motor de almacenamiento es un módulo de software que un sistema de administración de bases de datos utiliza para crear, leer y actualizar datos de una base de datos. Hay dos tipos de motores de almacenamiento en MySQL: transaccionales y no transaccionales.

Las tablas transaccionales tienen varias ventajas sobre las no transaccionales:

- Más seguras. Incluso si MySQL cae o tiene problemas de hardware, puede recuperar los datos, mediante recuperación automática o desde una copia de seguridad más el log de transacciones.
- Puede combinar varios comandos y aceptarlos todos al mismo tiempo con el comando COMMIT (si autocommit está desactivado).
- Puede ejecutar ROLLBACK para ignorar los cambios (si autocommit está desactivado).
- Si falla una actualización, todos los cambios se deshacen. (Con tablas no transaccionales, todos los cambios son permanentes.)
- Motores de almacenamiento transaccionales pueden proporcionar mejor concurrencia para tablas que tienen varias actualizaciones concurrentes con lecturas.

Motores de almacenamiento compatibles con MySQL:

- InnoDB
- MyISAM
- Memory
- CSV
- Merge
- Archive
- Federated
- Blackhole

MyISAM es el motor de almacenamiento original. Es un motor de almacenamiento rápido. No admite transacciones. MyISAM proporciona bloqueo a nivel de tabla. Se utiliza principalmente en web y almacenamiento de datos. El motor de almacenamiento de memoria crea tablas en la memoria.

El motor de almacenamiento de memoria es ideal para crear tablas temporales o búsquedas rápidas. Los datos se pierden cuando se reinicia la base de datos.

i) ¿Explique resumidamente de qué manera gestiona MySQL los espacios de tablas (tablespaces)?

Los tablespaces son áreas en disco donde se almacenan datos e índices de las bases de datos. Se gestionan principalmente a través del motor de almacenamiento InnoDB.

Gestión de tablespaces:

- **Tablespace general:** InnoDB utiliza un archivo compartido (ibdata1) donde guarda todas las tablas e índices, que crece dinámicamente según sea necesario.
- **Tablespaces individuales:** Con la opción innodb_file_per_table, cada tabla tiene su propio archivo .ibd, lo que facilita la gestión de almacenamiento.
- **General Tablespaces:** Introducidos en MySQL 5.7, permiten que varias tablas compartan un mismo tablespace, ofreciendo mayor flexibilidad.

Espacio libre y redimensionamiento:

- InnoDB gestiona el espacio usando páginas de 16 KB. El espacio liberado por registros eliminados puede ser reutilizado.

- Los tablespaces crecen automáticamente, pero reducir su tamaño requiere operaciones como OPTIMIZE TABLE.

j) ¿Dónde almacena físicamente las bases de datos MySQL?

Los datos, registros, estructuras de tablas y base de datos se almacenan en ficheros. Las distintas bases de datos se encuentran almacenadas en una estructura de directorios, correspondiendo cada uno a una base de datos.

En la mayoría de instalaciones de Linux en /var/lib/mysql. En Windows y otros sistemas operativos la estructura es la misma aunque cambie la ubicación de la instalación.

k) Enumere y describa brevemente qué tipos de índices implementa MySQL.

1. **Índice Primario:** Garantiza que los valores en la columna o conjunto de columnas sean únicos y no nulos. Se utiliza para acceder rápidamente a las filas por su clave primaria.
2. **Índice Único:** Asegura que los valores sean únicos, pero permite valores nulos. Se emplea para evitar duplicados en una columna o conjunto de columnas específicas, además de mejorar el rendimiento en búsquedas.
3. **Índice Regular:** No impone restricciones de unicidad ni de nulidad. Se utiliza para acelerar las consultas en columnas donde no es necesario garantizar la unicidad de los datos.
4. **Índice Compuesto:** Cubre múltiples columnas para optimizar consultas que las involucren.
5. **Índice de Texto Completo:** Optimiza búsquedas en grandes volúmenes de texto. Ideal para búsquedas de palabras clave.
6. **Índice Espacial:** Optimiza consultas con datos geoespaciales. Para realizar consultas eficientes basadas en coordenadas o geometrías.
7. **Índice Hash:** Rápido para búsquedas por igualdad, pero no para rangos.
8. **Índice Inverso (Reverse Index):** Se utiliza en casos especiales donde el orden inverso de los datos optimiza las consultas.

I) Instale la herramienta MySQL Workbench. Esta instalación deberá realizarse en la máquina host, NO EN LA MV.

- Cree una nueva conexión al servidor MySQL instalado en c) y de acuerdo a lo configurado en d).

Obtener la dirección IP de la máquina virtual.

```
Seleccinar ubuntu@mv-sgbd: ~
C:\Users\criss>multipass shell mv-sgbd
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-118-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of mié 21 ago 2024 15:11:26 -03

System load:          0.63
Usage of /:           8.0% of 28.89GB
Memory usage:        15%
Swap usage:          0%
Processes:           135
Users logged in:      0
IPv4 address for enp0s3: 10.0.2.15
IPv4 address for enp0s8: 192.168.100.25
IPv6 address for enp0s8: 2803:9800:94c0:8fca:5054:ff:feb1:8b3c
```

Configurar MySQL Workbench

Connection Name: mv-sgbd

Connection Remote Management System Profile

Connection Method: Standard (TCP/IP) Method to use to connect to the RDB

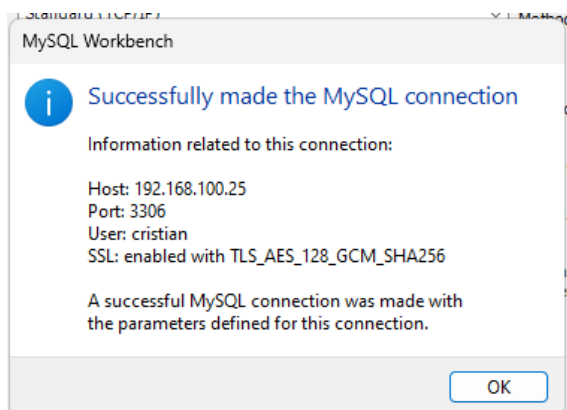
Parameters SSL Advanced

Hostname: 192.168.100.25 Port: 3306 Name or IP address of the server host - and TCP/IP port.

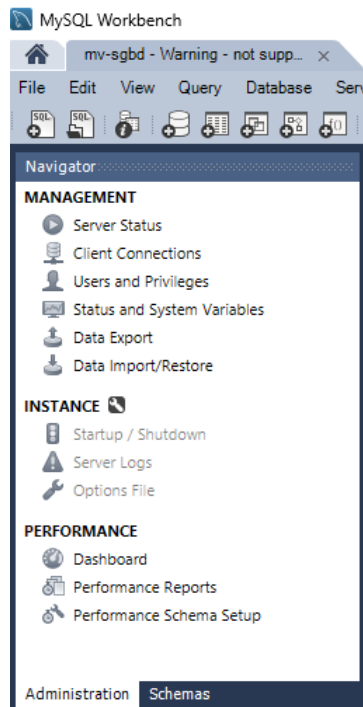
Username: cristian Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.



- Ingrese a la consola de administración de MySQL Workbench y describa las opciones y funcionalidades que observa.



- **MANAGEMENT**

- **Server Status**: Permite ver el estado actual del servidor MySQL, incluyendo estadísticas de rendimiento y otros detalles importantes como las conexiones activas y el tiempo de actividad.
- **Client Connections**: Muestra información detallada sobre las conexiones de los clientes al servidor, como los usuarios conectados, las consultas que están ejecutando, y el estado de las conexiones.
- **Users and Privileges**: Proporciona una interfaz para gestionar usuarios y sus privilegios en la base de datos. Aquí es posible crear, modificar o eliminar usuarios y ajustar sus permisos.
- **Status and System Variables**: Permite ver y modificar las variables de sistema de MySQL que afectan el comportamiento del servidor.
- **Data Export**: Facilita la exportación de bases de datos o tablas a diferentes formatos, como SQL o CSV.
- **Data Import/Restore**: Proporciona herramientas para importar datos o restaurar una base de datos desde un archivo de respaldo.

- **INSTANCE**

- **Startup / Shutdown**: Permite iniciar o detener el servidor MySQL.
- **Server Logs**: Muestra los registros del servidor, lo que es útil para monitorear errores, advertencias, y otra información relevante generada por MySQL.
- **Options File**: Permite editar el archivo de configuración de MySQL (my.cnf o my.ini), donde se pueden modificar opciones de configuración del servidor.

- **PERFORMANCE**

- **Dashboard**: Ofrece una vista gráfica del rendimiento del servidor, mostrando métricas como el uso de CPU, memoria, y consultas activas.
- **Performance Reports**: Genera informes detallados sobre el rendimiento del servidor, lo que es útil para identificar y resolver cuellos de botella.
- **Performance Schema Setup**: Permite configurar el esquema de rendimiento, una herramienta avanzada para monitorear el rendimiento de MySQL a nivel de operación.

- Investigue con la documentación oficial y elabore un resumen sobre:
 - Las funcionalidades provistas por MySQL Workbench:
 - <https://dev.mysql.com/doc/workbench/en/wb-intro.html>
 - <https://www.mysql.com/products/workbench/features.html>

Funcionalidades de MySQL Workbench

1. **Desarrollo de Bases de Datos:**
 - **Editor SQL Visual:** Permite crear, editar y gestionar consultas SQL de manera visual. Los usuarios pueden construir consultas complejas y ejecutar comandos SQL con facilidad.
 - **Gestión de Objetos:** Facilita la administración de tablas, vistas, índices, disparadores y otros objetos de la base de datos.
 - **Herramientas de Inspección:** Incluye herramientas para inspeccionar y editar esquemas y tablas, ofreciendo una vista detallada de la estructura y los datos de la base de datos.
2. **Administración de Bases de Datos:**
 - **Gestión de Usuarios:** Permite crear y gestionar cuentas de usuario y sus permisos, facilitando la administración de la seguridad de la base de datos.
 - **Monitoreo del Rendimiento:** Ofrece herramientas para el monitoreo del rendimiento del servidor MySQL, ayudando a identificar y solucionar problemas de manera proactiva.
3. **Diseño y Modelado de Bases de Datos:**
 - **Modelado EER (Entidad-Relación Extendida):** Proporciona un entorno gráfico para diseñar modelos de bases de datos, permitiendo la creación de diagramas EER que representan las relaciones entre las tablas y otros objetos.
 - **Ingeniería Directa e Inversa:** Facilita tanto la creación de bases de datos a partir de un modelo EER como la generación de un modelo a partir de una base de datos existente.
4. **Migración de Bases de Datos:**
 - **Asistente de Migración:** Herramienta que facilita la migración de bases de datos desde otros RDBMS (Sistemas de Gestión de Bases de Datos Relacionales) hacia MySQL, simplificando la conversión de estructuras de datos y procedimientos almacenados.
 - Objetos que se muestran en la consola de administración, por ejemplo: ¿Qué es una tabla?, ¿Qué es un índice? ¿Qué es una vista? ¿Qué es un disparador (trigger)? ¿Qué son las cuentas de usuario?, etc.

Objetos en la Base de Datos

- **Tabla:** Es un objeto fundamental en la base de datos que almacena datos en filas y columnas. Cada columna tiene un tipo de dato específico, y las tablas pueden relacionarse entre sí mediante claves foráneas.
- **Índice:** Mejora la velocidad de recuperación de datos en una tabla al permitir búsquedas más rápidas. Se puede crear sobre una o más columnas de una tabla y es esencial para optimizar el rendimiento de consultas.
- **Vista:** Consulta almacenada que actúa como una tabla virtual. Permite mostrar datos de una o varias tablas en una estructura personalizada, facilitando el acceso y la consulta de datos sin alterar las tablas subyacentes.
- **Disparador (Trigger):** Conjunto de instrucciones SQL que se ejecuta automáticamente en respuesta a ciertos eventos en una tabla, como inserciones, actualizaciones o eliminaciones. Se utiliza para mantener la integridad de los datos y automatizar procesos dentro de la base de datos.
- **Cuentas de Usuario:** Son los perfiles que tienen acceso a la base de datos. Cada cuenta puede tener permisos específicos, como la capacidad de leer, escribir, modificar o administrar la base de datos, asegurando que solo usuarios autorizados puedan realizar ciertas operaciones.

m) Cree una base de datos, que como único requisito obligatorio debe llamarse “SGBD2023”; para ésto investigue qué parámetros opcionales puede establecer y/o configurar para la creación. Escoja, y fundamente, los que considere adecuados.

Dentro de la máquina virtual, ingresar a mysql con el usuario y contraseña definidos previamente en el punto d).

```
Seleccinar ubuntu@mv-sgbd: ~
ubuntu@mv-sgbd:~$ mysql -u cristian -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.4.2 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Crear la base de datos.

```
Seleccinar ubuntu@mv-sgbd: ~
mysql> CREATE DATABASE SGBD2023
->   DEFAULT CHARACTER SET utf8mb4
->   DEFAULT COLLATE utf8mb4_general_ci;
Query OK, 1 row affected (0,02 sec)
```

- **CHARACTER SET: utf8mb4**
 - Es ideal para aplicaciones que manejan contenido multilingüe o internacional, ya que soporta la mayoría de los caracteres en cualquier idioma.
- **COLLATE: utf8mb4_general_ci**
 - Adecuada para bases de datos donde la velocidad de las consultas es una prioridad.
 - Insensibilidad a Mayúsculas/Minúsculas, lo que facilita las búsquedas y comparaciones de texto donde el caso no es relevante.
- **Motor de Almacenamiento (ENGINE): InnoDB**
 - Aunque no se especifica directamente en la creación de la base de datos, las tablas dentro de la base de datos se crean utilizando un motor de almacenamiento. **InnoDB**
 - Soporta transacciones que cumplen con las propiedades **ACID** (Atomicidad, Consistencia, Aislamiento, Durabilidad), lo que garantiza la integridad de los datos incluso en situaciones de fallos del sistema.
 - Soporta claves foráneas, lo que ayuda a mantener la integridad referencial entre tablas, evitando problemas como registros huérfanos.

A continuación cree en esa base de datos una tabla “Tabla1” con al menos tres campos de distinto tipo, investigando también sobre las opciones y configuraciones opcionales. Documente cada paso con una copia de pantalla del asistente de creación o el comando utilizado.

```

Seleccionar ubuntu@mv-sgbd: ~
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| SGBD2023 |
| information_schema |
| menagerie |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0,00 sec)

```

Ingresar a mysql con el usuario, contraseña y la base de datos específica donde se desea trabajar.

```

Seleccionar ubuntu@mv-sgbd: ~
ubuntu@mv-sgbd:~$ mysql -u cristian -p SGBD2023
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.4.2 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Creando la tabla **Tabla1**

```

Seleccionar ubuntu@mv-sgbd: ~
mysql> CREATE TABLE Tabla1 (
-> id INT AUTO_INCREMENT PRIMARY KEY,
-> nombre VARCHAR(30) NOT NULL,
-> fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP);
Query OK, 0 rows affected (0,06 sec)

mysql>

```

Parámetros Utilizados

- **id INT AUTO_INCREMENT PRIMARY KEY**
 - **INT**: el campo es de tipo entero.
 - **AUTO_INCREMENT**: el campo incrementa automáticamente su valor en 1 cada vez que se inserta un nuevo registro.
 - **PRIMARY KEY**: este campo será la clave primaria de la tabla, lo que significa que será único y no permitirá valores nulos.
- **nombre VARCHAR(100) NOT NULL**
 - **VARCHAR(30)**: campo de tipo cadena de texto con un máximo de 30 caracteres.

- **NOT NULL:** Establece que este campo es obligatorio, es decir, no puede tener valores nulos.
- **fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP**
 - **DATETIME:** Un campo que almacenará tanto la fecha como la hora.
 - **DEFAULT CURRENT_TIMESTAMP:** Configura el campo para que, por defecto, tome el valor de la fecha y hora actuales en el momento en que se inserta un nuevo registro.

```

mysql> DESCRIBE Tabla1;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int           | NO   | PRI | NULL             | auto_increment |
| nombre        | varchar(30)   | NO   |     | NULL             |                |
| fecha_creacion | datetime      | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql> _

```

n) Elija al menos cinco conceptos vistos en la clase de teoría de “Introducción a las bases de datos” e identifique y describa brevemente cómo son implementados por MySQL.

1. La **Independencia Lógica** se refiere a la capacidad de cambiar el esquema lógico de la BD (tablas y relaciones) sin afectar a las aplicaciones que dependen de esos datos.
2. En **MySQL** se refiere a las **vistas**. Permite crear **vistas** que proporcionan una forma de representar los datos, abstrayendo detalles de la estructura física.
3. La **Independencia Física** se refiere a la capacidad de cambiar el esquema físico (tipos de archivos, índices, sist. de almacenamiento) sin afectar el esquema lógico.
4. **MySQL** tiene **motores de almacenamiento**, como InnoDB o MyISAM, que manejan el almacenamiento y la recuperación de los datos. Algún cambio en ellos no se ve reflejado en el esquema lógico.
5. La **Arquitectura de MySQL** sigue el modelo cliente/servidor, donde el servidor gestiona la BD, procesa las consultas y maneja la administración de datos y las transacciones, el cliente puede ser una interfaz de línea de comandos, una aplicación que usa un conector o herramientas gráficas, interactúa con el servidor para realizar consultas y otras operaciones. **MySQL** no se adhiere estrictamente al **modelo de Arquitectura ANSI/SPARC**, pero puede entenderse en el contexto de estos niveles:
 - a. **Nivel Externo** se refiere a las vistas que la BD proporciona a los usuarios.
 - b. **Nivel Conceptual** se especifica a la creación de tablas, definición de sus relaciones a través de claves primarias y foráneas.
 - c. **Nivel Interno** se refiere a los motores de almacenamiento.